Cider PRESS

San Francisco Apple Core

IN THIS ISSUE:

		PAGE:
Active Volunteers & Committees		4
Cider Press Policy		4
Exec Apple		
President's Message	Peter Weiglin	5
General Meeting Notes	Bill Neff	5
Activity Schedule	Steve Lloyd	6
Marketing Dept. Notes	Don Scellato	7
Programming Contest	Don Scellato	7
B*A*S*I*C	Joyce Conklin	8
Review-Datafax	Joyce Conklin	9
Apple Core Offerings		11
Apple /// & Epson Printing	Jim Linhart	12
Children's Benefit Computer Fair	Allan B. Woods	14
Notes From the IAC		14
Cover Contest Winners	Gene Wilson	16
CP/M'83 - A Few Notes	Gene Wilson	17
SELECTOR V - A Review	Woody Liswood	18

GENERAL MEETINGS:

FEBRUARY 19

MARCH 19

APRIL 16

FEBRUARY, 1983

Cybet Computers (415) 933-1131

COMPUTERS FOR HOME AND BUSINESS

2270-I OAK GROVE ROAD
WALNUT CREEK, CA 94598
TEL: (415) 933-1131

MAIL ORDER DISCOUNTS

To all Apple Core Members **-20% Off-** Mail Order Only.

Beagle Bros.:

Apple Mechanic	$29.50
Alpha Plot	39.50
Beagle Bag	29.50
DOS Boss	24.00
Flex Text	29.50
Frame-Up	29.50
TypeFaces	20.00
Utility City	29.50

Penguin Software:

The Graphics Magician	$59.95
Apple Graphics Special	39.95
Complete Graphics Sub.II	69.95
Additional Fonts & Sets	19.95

Books:

Basic Prog. Guide Pascal	9.95
CP/M Primer	14.95
Beneath Apple DOS	$19.95
Bag of Tricks	36.00
Data Base Mgmnt.	16.95
Using 6502 Asmb. Lang.	19.95
Science & Eng. Prog.	15.99
Osborne CP/M User's Gd.	12.99
Practical Pascal Prog.	15.99
6502 Asmb. Lang. Subrout.	15.95
Apple II Users Guide	16.95
VisiCalc Home & Ofc. Com.	15.99
Asmb. Lang. Prog. f/ AII	12.95
Kids & the Apple	19.95
Intro. to Microcomputers	12.50
Some Common Basic Prog.	14.99

AND MORE!

Omega Microware:

Locksmith	$99.00
The Inspector	59.95
The Consolidator	49.95
The Manager	34.95
Watson	49.95

Educational:

Algebra 1	$39.95
Algebra 2	39.95
SAT Word Skills	49.00
CompuRead	29.95
Counting Bee	29.95
Decimals	49.00
Fractions	49.00
CompuSpell	29.95
Elementary Math	39.95
Spelling Bee/Reading	39.95
Typing Tutor	24.95
Master Type	39.95
MatheMagic	99.95
Magic Words	69.95
Magic Window	99.95
The Linguist	40.00
U. S. Const. Tutor	29.95

VisiCorp:

VisiCalc	$250.00
VisiFile	250.00
VisiPlot	200.00

Games:

Alien Ambush	$24.95
Snack Attack	29.95
Apple Panic	29.95
Space Vikings	49.95
Arcade Machine	59.95
Aztec Adventure	39.95
Bolo	39.95
Castle Wolenstein	29.95
Choplifter	34.95
Star Blazer	31.95
Cranston Manor	34.95
Story Machine	34.95
Crossfire	29.95
Davids Midnight Magic	34.95
Delta Drawing	59.95
FaceMaker	34.95
Frogger	34.95

AND MORE!

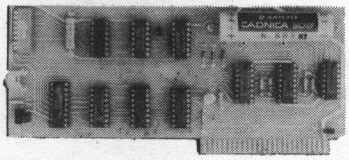
Give us a call for other fine **Apple** products that we carry.

APPLE PERIPHERALS ARE OUR ONLY BUSINESS

TIME II

THE MOST POWERFUL, EASIEST TO USE CLOCK FOR YOUR APPLE

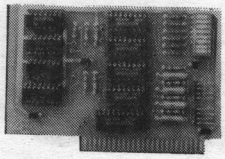
- Time in hours, minutes and seconds.
- Date with year, month, day of week and leap year.
- Will enhance programs for accounting, time and energy management, remote control of appliances, laboratory analysis, process control, and more.
- 24-hour military format or 12-hour with AM/PM indication.
- User selectable interrupts permit foreground/background operation of two programs simultaneously.
- Crystal controlled for .0005% accuracy.
- Easy programming in basic.
- On board battery backup power for over four months power off operation (battery charges when Apple is on).



- Twenty-seven page operating manual included with many examples of programs to use with your Apple in any configuration.
- Includes disk containing a DOS Dater and many other time oriented utilities plus over 25 user contributed programs at no extra cost.

PRICE $129.00

SUPER MUSIC SYNTHESIZER



- Complete 16 volce music synthesizer on one card. Just plug it into your Apple, connect the audio cable (supplied) to your stereo and boot the disk supplied and you are ready to input and play songs.
- It's easy to program music with our compose software. You will start right away at inputting your favorite songs. The Hi-Res screen shows what you have entered in standard sheet music format.
- We give you lots of software. In addition to Compose and Play programs, the disk is filled with songs ready to run.
- Easy to program in basic to generate complex sound effects.
- Four white noise generators which are great for sound effects.
- Plays music in true stereo as well as true discrete quadraphonic.
- Envelope control.
- Will play songs written for ALF synthesizer (ALF software will not take advantage of all the features of this board. Their software sounds the same in our synthesizer.)
- Automatic shutoff on power-up or if reset is pushed.
- Many many more features.

PRICE $159.00

ANALOG TO DIGITAL CONVERTER

- 8 Channels
- 8 Bit Resolution
- On Board Memory
- Ratiometric Capability
- Fast Conversion (.078 ms per channel)
- Eliminates The Need To Wait For A/D Conversion (just PEEK at data)
- A/D Process Totally Transparent to Apple (looks like memory)

The analog to digital conversion takes place on a continuous, channel sequencing basis. Data is automatically transferred to on board memory at the end of each conversion. No A/D converter could be easier to use.

Our A/D board comes standard with 0, 10V full scale inputs. These inputs can be changed by the user to 0, -10V, or -5V, +5V or other ranges as needed.

The user connector has + 12 and -12 volts on it so you can power your sensors. (These power sources can be turned off with on board dip switch).

Accuracy 0.3% Input Resistance 20K Ohms Typ

A few applications may include the monitoring of ● flow ● temperature ● humidity ● wind speed ● wind direction ● light intensity ● pressure ● RPM ● soil moisture and many more.

PRICE $129.00

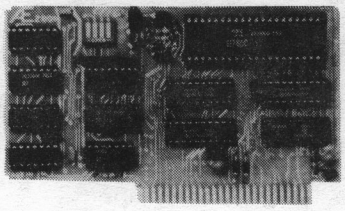
DIGITAL INPUT/OUTPUT BOARD

- Provides 8 buffered outputs to a standard 16 pin socket for standard dip ribbon cable connection.
- Power-up reset assures that all outputs are off when your Apple is first turned on.
- Features 8 inputs that can be driven from TTL logic or any 5 volt source.
- Your inputs can be anything from high speed logic to simple switches.
- Very simple to program, just PEEK at the data.
- 4 other outputs are also provided. User 1, reset, interrupt request, non-maskable interrupt.
- Now on one card, you can have 8 digital outputs and 8 digital inputs each with its own connector. The super input/output board is your best choice for any control application.

PRICE $62.00

Z-80 CARD

- TOTALLY compatible with all CP/M software.
- Executes the full Z-80 and 8080 instruction set.
- Allows you to run your Apple CP/M based programs.
- Does EVERYTHING the other Z-80 boards do, plus supports Z80 Interrupts.
- Hardware and software settable switch options.
- An on-card PROM eliminates many I.C.'s for a cooler, less power consuming board.
- Complete documentation included. (user must furnish software)



PRICE $139.00

Since our inception, Applied Engineering has continually expanded its line of Apple peripherals bringing you easy-to-use designs.

We are the innovators not the imitators. Utilizing state-of-the-art technologies. Applied Engineering is continually improving its products. The above represents our most recent development. Applied Engineering offers you the highest quality peripherals at the lowest possible price.

Applied Engineering's products are fully tested with complete documentation and available for immediate delivery. All products are guaranteed with a one year warranty.

All Orders Shipped Same Day.
Texas Residents Add 5% Sales Tax.
Add $10.00 If Outside U.S.A.

Send Check or Money Order to:
APPLIED ENGINEERING
P.O. Box 470301
Dallas, TX 75247

See Your Dealer
or Call (214) 492-2027
7 Days a Week
Master Card & Visa Welcome

ACTIVE VOLUNTEERS & COMMITTEES

CIDER PRESS

OFFICERS:

President:	Peter C. Weiglin
Vice-President:	Rance Fields
Vice President:	A.S. ("Woody") Liswood
Secretary:	Glenda Lloyd
Treasurer:	Gary Knox

DIRECTORS:

Alan L Smerican	415/566-2342
Peter C. Weiglin	415/345-3980
Rance Fields	415/563-2173
Kenneth D. Silverman	408/246-5449
Stephen C. Lloyd	415/222-5868
A.S. ("Woody") Liswood	415/585-6375
Gene Wilson	707/996-8612
Don Scellato	415/672-2051

COMMITTEES:

Membership -		
Alan L Smerican		415/566-2342
	Ted Chatterton	
Cider Press -		
Editor	Gene Wilson	707/996-8612
Asst. Ed.	Peter Weiglin	415/345-3980
Asst. Ed.	Ken Silverman	408/246-5449
Graphics	Jim Linhart	415/652-0909
BASIC	Joyce Conklin	415/341-6955
Adv. Mgr.	Alan W. Johnson	415/479-1099
	Richard Highsmith	415/871-2679
Int. Grp. Coord.	Steve Lloyd	415/222-5868
Marketing Dir.	Don Scellato	415/672-2051
Review Group	("Woody") Liswood	415/585-6375
Demo Coord'r	Rance Fields	415/563-2173
IAC Rep.	Peter C. Weiglin	415/345-3980

CIDER PRESS POLICY

All manuscripts, photographs, and other materials will be considered submitted free and released for publication. They become the property of the San Francisco Apple Core and the CIDER PRESS. Authors should clearly mark all material submitted for publication so that proper credit can be given.

The views expressed in any articles are those of the authors, and not necessarily those of the San Francisco Apple Core, Inc., the Editor, Club Officers or Directors . . . or anyone else in their right minds.

The San Francisco Apple Core Incorporated is a non - profit organization comprised of and supported by owners and users of Apple I I and Apple / / / computers. The Club is run by officers and committees, and the Club endeavors to aid Apple owners through its Program Libraries, Program Disks, meetings, and this publication.

Article Submission: The CIDER PRESS welcomes any and all articles dealing with Apple computers and their associated hardware and software. Articles may be submitted (typed or printed, double - space), or as disk text files using any of the popular word processors or text editors. Any author who has an article published in the CIDER PRESS will receive a free copy of the most current Disk - of - the - Month.

Reprints: All articles appearing in the CIDER PRESS not copyrighted by the author may be reprinted by other *non - profit* Apple User Groups, without prior permission, so long as proper credit is given to both the author and the CIDER PRESS. Permission to reprint a copyrighted article must be obtained in advance, by writing to the authors, c/o the San Francisco Apple Core.

San Francisco Apple Core
1515 Sloat Blvd.
San Francisco, CA 94132

Exec Apple

President's Message

Peter C. Weiglin

The San Francisco Apple Core's January meeting was one of those that makes the whole idea of the Apple Core worth while. What the 200 or so who attended saw was a first look at the Apple //e. Then, a towel was whisked from another computer-sized lump, and we beheld (gasp!) LISA!. We were also treated to a multimedia show via videotape. Thanks to members Harlan Felt, Joe Budge, Pat Caffrey for making that one happen. Particularly Harlan Felt.

February's meeting will be another Swap Meet, at which all kinds of stuff will be transferred between members. Reminds me of the Souks in Riyadh, and it's just as much fun.

Meanwhile, the Wilsonian Cider Press is still going strong; and the Marketing efforts, under Don Scellato, are really humming. The programming contest (see article in this issue) is another way of encouraging you to participate in Club activities.

When I think of where we were a year ago, bogged down in organizational, procedural, club/political and conceptual/theoretical conversation about the future of the Club . . . well, we have been taking action. I won't pretend that no mistakes have been made, but we're moving. The next area that needs attention is the interest groups and educational activities, and that's under way. If you'ld like to help, or want more information, call Steve Lloyd (415) 571-7370 or Joyce Conklin (415) 341-6955.

In short, the SFAC is what it always has been: the sum of its volunteers' efforts. This is not an exclusive group; anyone can join in. You are hereby invited to do so.

One more thing: the CP/M '83 show in San Francisco in January was alleged to contain a seminar/meeting on Apple CP/M to be conducted by the International Apple Core, of which the SFAC is a member. This seems to have been misrepresentation by the show management, as no arrangements were made with the IAC or SFAC. But then, they also showed Apple Computer as an exhibitor, which wasn't the case. Apple says they never intended to exhibit at that show.

We also hear that the "Applefests" now being advertised for 1983 may be squelched; something about unauthorized use of Apple's trademarks. So before you commit, hang on. We'll get the story as soon as it's available.

SAN FRANCISCO APPLE CORE
General Meeting
Ft. Mason January 22, 1983

Peter called us to some sort of order about 5 after 10 with a few announcements.

1. Finalizing the slate for the BOD election. (See elsewhere in this issue for the list of candidates.)
2. Don asked for volunteers for the February swap meet and for the March Computer Faire. The club will sell your stuff for 5% commission. He also called our attention to the new 20% discount service on software.
3. Ken offered a special discount to club members on the IAC Apple Orchard. The current issue tells all about the new APPLE IIe computer. Ken is in charge of booth registration for the swap meet.
4. Joyce needs articles for the BASIC column.
5. Gene was too tired to announce that the Cider Press always needs articles.

Rance (nee Randy) presents: Harlan - the IIe and LISA. The wonderful Glossary for Apple Writer and Epson. Incredible Jack the do-it-all disk: WP, DB, ARITHMATIC. PFS new super graphics program. Silicon Vally Systems: Word Handler. The APPLE IIe is, of course the latest APPLE II and replaces the II and II+. It is about 95% compatible with present soft, firm, and hardware. Most problems will be easily solved. It is or has: only 31 chips, 64K RAM, 16K ROM, MMU, IOU, full curser, delete, standard key arrangement, all repeat, game switches on keyboard, lower case, CAP lock, dual purpose shift, back with DB slots, screwdown lid (genius always seems so obvious, "why didn't we think of that before..."), very low power drain, socket for number pad, new manuals. Price: IIe 1395, 80 col. card 125, 80 col. card with 64K RAM 295, package: IIe, 80 col, disk II, monitor III = 1995!!! quick questions: PASCAL OK, VISICALC OK (new version coming for extended memory), ditto Applewriter 2, new DOS????, double graphic resolution, same bugs in AppleSoft, the APPLE II and II+ are relects, softswitch readout, sound same, auto repeat may interfere with some music programs.

LISA is the personal business executive computer which requires programming ability at about the level of building a toy block house! Its software has all the standard business programs but requires only menu function selection to use them. The menus are accessed by the "MOUSE" controled curser. Touch what you want or want to do. The 68000 CPU computer is attached by spooler to graphics and hard printers, only 9995.

THIS WAS THE FIRST TIME LISA HAS BEEN OUT OF THE APPLE HOUSE!!! question: what will it do? YES!!! 12:36

Bill

MONTHLY SCHEDULE OF ACTIVITIES

The monthly meeting of the San Francisco Apple Core will be held Saturday February 19, 1982. It will be held in building C at Fort Mason, San Francisco.

Interest Group Schedule for the Month of February

Monday	Tuesday	Wednesday	Thursday
	1 FORTH East Bay	2	3 Green Apples
7	8 Visicalc CP/M	9	10 BASIC
14 FORTH	15 Directors	16 Apple ///	17 Green Apples
21 Assembly	22 Epson CP/M	23	24 Green Apples

The monthly meeting of the San Francisco Apple Core will be held Saturday March 19, 1983. It will be held in building C at Fort Mason, San Francisco.

Interest Group Schedule for the Month of March

Monday	Tuesday	Wednesday	Thursday
	1 FORTH East Bay	2	3 Green Apples
7	8 VisiCalc CP/M	9	10 BASIC
14 FORTH	15 Directors	16 Apple ///	17 Green Apples
21 Assembly	22 Epson CP/M	23	24 Green Apples
28	29	30	31 Green Apples

Interest Group Leaders

Apple ///	Don Norris	393-6290 day
Assembly	John Lynch	665-0136
Board of Directors	Peter Weiglin	345-3980
CP/M	Jim Ayers	383-0473 day
		383-5759 eve
Epson	Mike Fisher	557-1250 day
		933-4435 eve
Forth	Bob Ackerman	626-6295
East Bay (formerly Graphics)	Molly Mitchell	526-9541
Green Apples BASIC	Gary Owens	521-4002
Intermediate BASIC	Gary Owens	521-4002
Product Review	Woody Liswood	585-6375
VisiCalc	Don Scellato	271-8895 day
		672-2051 eve

Echos From The Past

Blaise Paskal has recently been getting lots of phone calls asking 'Why?' and 'How do I get around this?'. Two of the most common are:

1) How do I get around the 'stack overflow' error when I'm compiling a program?

2) Is there a keypress routine that works?

Answering the first question is rather simple, turn swapping on. This is done by placing the compiler option, (*$s+*), as the very first line of your program file. It goes before the program heading as illustrated below.

```
(*s+*)
program Name;

begin
end.
```

If this still doesn't help, try the (*s++*) option.

The manual explains that the swapping option provides more space for the symbol table. What it means is that the compiler has run out of space to keep track of things like variables and procedure names. Turning the swapping option on increases the available space.

Don't worry about the compilation speed being slower, once your program is compiled the swapping option will have no effect on its execution speed.

The second question is a little harder to answer and requires some explaination. The Pascal (sp) system that runs on the Apple has a type-ahead buffer. This is a few bytes in memory where characters from the keyboard are temporarily held until the program is ready for them. When the system moves a character from the keyboard to the type-ahead buffer, it also moves the indication that a character is available.

Some console devices (things plugged into slot 3), such as older VIDEX 80 column cards disable the type-ahead buffer. This means that the character available indication can only be obtained from the keyboard, not the type-ahead buffer. Keypress routines that only look at the type-ahead buffer won't work for these devices.

Keypress routines that only look at the keyboard won't always work with devices that do use the type-ahead buffer, such as the newer VIDEX 80 column cards. This is because the system regularly moves characters and the character available indication from the keyboard to the type-ahead buffer. This can cause a false keypress indication even though there are characters available in the buffer.

When you are writing programs that must work with both types of devices, you have to use a keypress routine that checks for both types of character available indications. The Keypress routine in the Applestuff unit only looks at the type-ahead buffer, it doesn't work with the older VIDEX cards.

The Keypress function shown in listing 1 below looks at both the type-ahead buffer and the keyboard, it should work with anything.

```
function Keypress : boolean;

type    Peeker = record case boolean of
                   true : (Address : integer);
                   false : (Contents : ©integer);
                   end;

var     Buffer_Read, Buffer_Write, C000 : Peeker;

begin
Buffer_Read.Address := -16616;
Buffer_Write.Address := -16615;
C000.Address := -16384;

(* When the keyboard strobe (bit 8) is high, a
character is available. *)
(* When the read pointer is different from the
write pointer, a character is available.*)

Keypress := ((C000.Contents mod 256)>128)
                        or
            ((Buffer_Read.Contents mod 256)
                        <>
            (Buffer_Write.Contents mod 256));
end;
```

Listing 1

Marketing Department Notes

Don Scellato

The Marketing Department is preparing a new general use San Francisco Apple Core disk for sale to Club members. This disk will feature a nember of recently contributed programs but due to the nature of two of the programs will not contain a large number. Two very exciting programs take up most of the space on the disk. One is a graphics game written on The Graphics Magician. The other is a "slide show" type of program which shows off several graphics screens.

We are also preparing a Graphics Applications Disk for sale to members and have just started a Small Business Applications Pak. We hope to have the Graphics Disk ready for release in March and the Small Business Pak ready for May.

The Marketing Department has set up with a major distributor of softwware and hardware to make these products available to Club Members at about 20% under retail price. The purpose of this arrangement is twofold. It will get commercial programs out to Club Members at bargain prices. It will also provide a source of revenue for the Club to maintain its activities. No individual or business will receive any proceeds from these sales. Any proceeds will go directly into the Club's treasury. Look for catalogs at the Club's regular meetings. They contain a limited listing of products we can obtain for you at reduced prices. If you want to place an order, fill out an Order Blank and leave it at the Sales Table. We only ask that you pay in advance in full for any products you wish to order. We'll place the order on the first working day after the meeting. It should take a few days to receive an order. We'll hold it for you for the next meeting. If you would rather have the product shipped to you instead of waiting, we can arrange the shipment for an additional $2.00 charge. As with other products sold at the Sales Table, California State Sales Tax must be charged. If you need something that isn't on the list, just contact Don Scellato or Joe Bianco at the meeting and we'll be glad to quote a price on the product you're looking for. Help the Club and help yourself by using this discount plan.

PROGRAMMING CONTEST

By

Don Scellato

The San Francisco Apple Core plans to run a programming contest which will begin this spring and will be open for three months to any programs which Club members wish to enter. The Club is arranging with hardware and software manufacturers to have them act as Contest Sponsors by contributing their products to the Club for use as prizes to be awarded to the winners of the Contest. Programs will be judged on the basis of originality, user friendliness, documentation, error-trapping, screen formatting, and similar criteria.

The Contest categories will be Home Applications, Management Applications, Games and Entertainment, Education, Programming Utilities, Graphics and Sound Applications, and Spreadsheet Applications. A formal announcement of the opening and closing dates of the Contest will be announced in a future issue of The Cider Press. At that time all of the rules and particulars of the Contest will be provided. The prizes will also be announced at that time.

The Club hopes to encourage each of those members who have taken the time to write a useful or entertaining program to enter that program in the Contest. All programs which are entered will become the property of the Club and will be released as Public Domain software.

LOOK FOR THE PARTICULARS IN FUTURE ISSUES OF THE CIDER PRESS AND LISTEN FOR ANNOUNCEMENTS AT THE NEXT TWO CLUB MEETINGS.

(c) Copyright 1983 by Joyce Conklin

The subject of Calls, which was promised last month, does not justify a column all by itself for those who are reading a column on BASIC. To know exactly how each Call works requires a knowledge of machine language, or at least Assembly language. If you have that, why are you reading this column? If not, just take my word for it; what a Call does is to access a machine language sub-routine built into the ROMs on your motherboard. It does what?!?! I hear you say ... well, all that means is that some of the most commonly performed functions have been built-in to the circuits inside your Rome Beauty, making it easier for you to code your programs. The listing of Calls which was published in the Best of the Cider Press for 1978-79 appears to have been written especially for Integer BASIC, which almost no-one uses any more. I will have to do some more investigation, but if I find a good list, it may be forthcoming in a future issue. For now, some of the most useful calls are as follows:

CALL -958	Clear screen from cursor position to the bottom of the screen
CALL -936	Clear the text window (I'll bet you thought "HOME" was the way to do that!)
CALL -868	Clear line from cursor to the right edge of line
CALL -756	Wait for keystroke (like the GET command)
CALL -151	Enter the monitor (no, no, that does not mean get on a Civil War naval vessel!)

A number of other CALLs that were used in Integer can be accessed directly by using Control-characters; for example, CALL -922 is the same as Ctrl-J, or line feed, CALL -926 is the same as <Return>, CALL -1052 is the same as Ctrl-G, etc. Play around with these sometime in your programming - if nothing else, you will find that there are easier ways to do these with Applesoft! The advantage of using these built-in routines is the added speed of machine language.

On to a new topic.... When you have a "whole bunch" of information, most of the time it stays in a pile (mine is neatly (???!) in cardboard cartons!) but is isn't extremely useful. To understand what your information means, you need to organize it, often needing to work with pairs or sets of data. I suppose you could do it with a collection of different variables, but you soon run out of distinct variable names if you have more than just a few pieces of data. This is where arrays and subscripted variables come in.

An array is a set of related items. In BASIC you can think of an array as a series of "mailboxes" or addresses. If all the iems to be identified are numbers, the array will be named as a numeric variable, e.g. A, A1, AR, ARTIFACT, etc. If any of the items is to contain cahracters other than numbers, the array must be a string array and named as a string variable, e.g. A$, A1$, AR$, or ARTIFACT$. So far, nothing much new, eh? But a variable has one element, and array has more than one. So any individual element of an array must be identified by the subscript. Suppose you have a numberical array of seven different elements. The array would be named, let's say, N. The individual elements would be N(1), N(2), N(3), . . . N(7). Notice that the designation is N(3), not N3. N(3) is one element of an arry, N3 is an individual variable.

The next aspect of using arrays is the DIM statement. Does that mean you're not too intelligent? No way - you wouldin't be reading this column if that were the case! DIM stands for dimension. Applesoft automatically sets aside space in memory for any arrays containing a subscripted value of 10 or less; this means that the array may contain up to 11 items, as the first item in an array is subscripted zero, N(0). For any array which contains, or could contain, during the course of RUNning the program, more than this number must be dimensioned before it is used. Failure to dimension an array, usually at the beginning of the program, will result in a "SUBSCRIPT OUT OF BOUNDS" error message. The format for this statement would be:

```
linum  DIM N(xxx)
```

where xxx represents the number of items in your array. You can have quite a few items, but if you get carried away, you may run out of memory. I ran a program yesterday with three different arrays, all subscripted to 700 and didn't run out of my 64K of memory, so you can see that there is quite a bit of space available in your Apple.

So....? What do you do with an array such as this? How about a program to score a multiple-choice test? In the following example, the array contains the set of correct answers to which each of the student answers is compared in obtaining the student score.

```
10 DIM K(20) : REM Key of correct answers
20 FOR L = 1 TO 20
30 PRINT "ANSWER TO QUESTION #";L;:INPUT K(L)
40 NEXT L
50 HOME
60 INPUT "HOW MANY STUDENTS";N
70 FOR LOOP = 1 TO N
80 INPUT "STUDENT NAME";N$
90 C=0
```

```
100 FOR IL = 1 TO 20
110 PRINT "STUDENT ANSWER TO QUES. #";IL;:INPUT A
120 IF A = K(IL) THEN C = C+1
130 NEXT IL
140 PRINT "FOR ";N$
150 PRINT "THE SCORE IS ";C;" OUT OF 20 OR"
160 PRINT C/20*100;"%"
170 GOSUB 5000
180 NEXT LOOP
199 END
4999 REM Screen Pause Subroutine
5000 VTAB 23:HTAB 8:PRINT"PRESS SPACE TO CONTINUE"
5010 GET AA$:IF AA$ = CHR$(32) THEN HOME:RETURN
5020 CALL -868:GOTO 5010
```

This program is designed to work entirely on the screen. Could you modify it so that it would print out lines 140-160 on your printer? Any one (you, or an aide) could enter the information and get the results. Line 10 DIMensions the array of correct answers; line 10-40 loads the array. Lines 70-130 "grade" the test. The variable C is the number of correct answers for each student; therefore it must be reset to zero for each new student (line 90). Notice that in line 120 the student answer is compared to the key, but the subscript is not the same one we used in lines 20-40. That doesn't matter; what you are concerned with is whether the student answer to question #1 (or whatever) is the same as the key answer to question #1. So you may use different variable names in the subscript to get the same element from the array.

A one-dimensional array, which is what we have been using so far, is not the limit of your Apple. You may also have two-dimensional arrays. For what, you ask? How about playing poker, or bridge or black-jack, or whatever you want. In keeping track of cards in a deck, you have two variables for each card, the rank and the suit. It is handy to think of the deck being represented by a two-dimensional array of 13 by 4 elements. Then if you want to "deal" a hand for 5-card draw, you would generated a randomly selected set of 5 different elements from your array. You need to have an array, rather than just the random selection of rank and suit, to prevent the same card from coming up more than once in a hand. It might be that you would want to set up a NEW game which says that 4 Kings of Clubs beats a royal flush in hearts, but that isn't what we started out to do!. The format for your DIMension statement (you must ALWAYS DIMension two-dimensional arrays) is:

```
linum DIM D(13,4)
```

All you have to do is to remember that the rank of the card must come first, and the suit second. Otherwise, you would have to have DIM D(4,13) if you wanted it the other way. How about working on this for the next month and I'll come up with mine. Then we can compare techniques.

If you have any topics which you would like to see addressed in this column, I would appreciate a note, care of the SFAC office address.

CIDER PRESS REVIEWS

DATAFAX

Review (c) Copyright by Joyce Conklin 1983

Distributed by: Link Systems
1640 19th St.
Santa Monica, CA 90404

Price: $199

System requirements: Apple II+ with 64K,
2 disk drives, DOS 3.3

The logo character for this database program is a sort of benign monster - probably the monster of disorganization with which we are all burdened! But almost as soon as you open this package, you will realize that the monster CAN be tamed...Hooray, hooray!... albeit not without a small bit of work on your part. Can anything which comes out of a building with stained glass windows, and where the missing steeple was sacrificed to obtain a more immediate view of the heavens be bad? Probably, but this package certainly isn't it! Before you even boot the disk you will realize what a tremendous amount of work went into designing this program AND its documentation. The documentation not only has a table of contents, it HAS AN INDEX! There is also a Command Tree reference card, which is useful because the program is set up in a series of levels which match the reference card.

So what does this wonder of wonders do? It is a data base, but unlike many of the others (PFS File, DB Master, etc.) it is free-form and allows you to enter text in any way that is convenient for you. I was originally attracted to it because I wanted to file word problems for my science classes and be able to retrieve them by subject. It will do this handily but that isn't all. If you absolutely must have repetitive organization in your files (for example, in filing books you would probably want the author's name, then the title, etc.) you can do this simply by entering the items on different lines. If you are SO disorganized that you need titles for each of these fields, maybe you'd better start with File Cabinet from the Apple Core Library, but it won't do what DataFax can do.

The very first thing you do is back up your disk. That's right - not send $50.00 (or even $5.00) for a backup - but do it yourself, right on a regular, unprotected disk. The preliminary documentation made comment on this (then) unusual state of affairs and appealed to your sense of fair play in not passing on the copyable material. The updated documentation makes no such comment, assuming that people who need applications software such as this will be willing to abide by the conventions and laws of the U.S. Once you have made your back-up copies (make two, they're cheap!) you can put the original disk away where it won't be harmed. This

back-up program disk should be left un-write-protected, as each time you boot, you can write the date to the disk, showing when you last worked with your database. The first section of the manual is a 67 page tutorial, which explains all the basic aspects of DataFax. There is even a demonstration database on the original program disk, which you can play with to get used to the way of moving through the various levels.

DataFax operates in run-time Pascal. You do not need a Pascal card in your Apple, but you do need to leave the program disk in the drive, as the program is too large to fit in memory all at one time. There is a message which may appear on the screen saying:

> You may remove the program disk
> Press <SPACE> to continue

This message appears during the process of making your back-up copy. However, if you follow the instructions you won't be able to back it up!! It won't cause the program to bomb, but it's a little difficult to copy something that isn't there.

The program is sensitive to variations in disk drive speed. I had made my back-up copies and took one of those with me to do a demonstration for some teachers. Uh-uh. The disk would'+ boot. (Not TOO embarassing!) When I got home I tried my back-up copy and sure enough it booted right away. A further check showed that my disk drive was running fast. I'll make another back-up when I get the drive speed adjusted.

The data base is conceived of as a series of file folders - the manilla kind you are all used to - and pages within these folders. Just as it is not totally productive to keep stuffing more papers in the same manilla folder in your file drawer, it is not a good idea to have more than a few "pages" in a "folder" of DataFax, although the theoretical limit is 255. After making your back-up, it is time to Open your file. All movement from one level to another is done with single key-strokes; it is not necessary to press <RETURN> - but many levels have default values which you may access by just pressing <RETURN>. If you are starting fresh, the next command would be N)ew; you also use this for creating each "folder", even if the subject matter is the same. For example, each different book you are filing would be in a N)ew folder. The subject matter can be thought of as a file drawer with each different item as a folder. If the "item" is large, there may be 2-3 pages in the folder. After you have entered your information, the next step is A)dd - keys. You cannot save a file without having at least one key by which to retrieve it at a later date. You may have many keys or few, but you must have one. It may be as short as one character; the upper limit is 28 characters. The shorter you make your keys the easier they are to type in and the more room there will be for data. You can key by date as well as by word, which would allow you have your file of correspondence sorted by date, if that is the way it would be most useful to you.

There is an edit menu so that you may change folders later on. Most of these edit functions are accessed using Ctrl- characters. There are installed keys now, but if you don't like the ones they chose (Ctrl-J always means line feed to you, instead of Ctrl-L) then you may change these default values and save the new set to disk.

The power of a data base is the ability to search for your data, and the search functions of DataFax are indeed powerful. In addition to being able to search using a specified keyword, you may also search for wild cards (e.g. AP= gets all folders with keys starting with AP), use AND and/or OR to combine your search, and you may search by a date OR a range of dates (e.g. all the folders with a key of 09/= or September). It even allows you to retrieve the folders found by the last search conducted! If you just like to look at a mass of data, you can retrieve ALL the folders in a file!!

You may sort your folders by using any of the keys mentioned above, and then by specifying the range, the folders found will be arranged. For example, if you had a data base of music, and wanted to sort your Mozart by K. number, you would SORT <Mozart> BY <K.=>; this last would sort all keywords with a K. as the first two characters. The sort function is the least speedy of the commands available, so don't try to do a sort while you're waiting to get the kids in the car to go to the Golden Arches!!

This is such a positive review that I would be ashamed to print it if it weren't all true. Sometimes you have to figure out a way to say something - anything - nice about a product, but in this case I haven't found anything BUT nice to say! It will take you a little time to become familiar with the commands, but all the default and selection values appear on the screen (at the top or bottom, depending upon which level you are working with) so you can be reminded. You can always work backwards to the main menu by using the <ESC> key. Set aside some time to organize your data, and you will never again find yourself in the position of buying something by mail because it sounds good, only to discover you have two or three of the things already! DataFax can really tame your data monster. Now, if only it did windows!!

APPLE CORE OFFERINGS

NOTES

The Disk Of the Month is a group of recently donated programs. Future issues of the Cider Press will alert you to new items for sale.

The "Best of Cider Press Disk, Vol xx" contains most of the programs printed in the publication "The Best of the Cider Press, Vol xx".

Cattlecar Galactica" Disk is for the experienced programmer only. It was written by Bruce Tognazzini; the program takes over your Apple II. The object is to get control back without turning off the Apple! (This is not as easy as it sounds! Typing in commands from many languages produces strange results. Attempting to list this program leads to hysterics!)

Library #1 contains over 340 programs on 20 diskettes. It is packaged in diskette holders and is bound in a SF Apple Core binder with over 30 pages of documentation. The documentation alone is available for three dollars.

Library #2 continues the tradition started with #1, and contains over 250 programs from the 1980 DOMs. This library also contains some special disks. Cattlecar Galactica, Apple II Plus Disk (for Apple II+ owners wishing to run 'Integer' programs in RAM memory), the new 16 Sector Utility Disk with all the Lawrence Hall of Science programs we used in the 13 sector (3.1, 3.2 and 3.2.1) disks, and the IAC disk of HiRes Games. The library includes disks, binder, plastic holders, and documentation. As with Library #1, the documentation is available alone for three dollars!

The Best of the Cider Press, 1978-1979 is a two year anthology of articles, program listings and charts from the SF Apple Core's newsletter, the "Cider Press". Sections range from articles for the beginner to charts of PEEKS, POKES, and CALLS.

The Best of the Cider Press, 1980 condenses the entire 1980 year into 116 pages of information on everything from a Beginner's Corner to Graphics and Shape Tables. A large Pascal section with many program examples is included.

Instructions: The easiest way to get these products is to visit the 'sales table' during any General Meeting. Otherwise, circle the items you wish to purchase, place this sheet into an envelope with check or money order (DO NOT send cash!) to:

San Francisco Apple Core, Inc.
1515 Sloat Blvd., Suite #2
San Francisco, CA 94132
Attn:Marketing Dept.

PRODUCT	MTG. PRICE	MAIL-USA
Library #1	98.00	101.50
Library #2	98.00	101.50
IAC Disks		
0-10	5.00	7.00
Pascal		
Attach		
Vol 3A		
3B		
Aug 80		
Oct 80		
Dec 80	5.00	7.00
Best of 80		
Feb 81		
Apr 81		
Aug 81		
Dec 81		
DOMs		
Mar 82		
Apr 82		
May 82	5.00	7.00
Cattlecar		
Best of 79		
Best of 80		
1980 Utility		
From West Coast Computer Faire		
Home Pac (3 disks)		
Basic Tutor "	20.00	25.00
Games Pac "		
Super Utility (1 disk)		
Pascal Sampler		
CP/M Sampler	10.00	12.00
Forth Sampler		
Silicon Gulch Games Fest		
Programming Aide		
Home & Hearth	5.00	7.00
Games Disks 1,2,3		
"Best of Cider Press"		
1978-79	3.00	4.50
1980		
"Cider Press"		
-subject to avail.	1.50	2.00
Pins		
PacMan		
SFAC	3.50	4.50
Apple		
Visicalc		
Templates	10.00	13.00

All foreign orders: Add 25% to above prices.

APPLE /// & EPSON PRINTING
by Jim Linhart

I own an Apple /// computer with an Epson MX-100 printer connected by an Universal Parellel Card. The printer's really versatile and I like all the fonts the printer can use, my favorite being COMPRESSED. If left to itself, the Epson prints in NORMAL font, which, in my opinion, is fairly boring. COMPRESSED looks very nice and it's the one I like to use when printing out letters, disk catalogs, and program listings. The only difficulty in getting into any font other than NORMAL is typing in a bunch of parameters, hoping that I didn't make any typing errors.

To save the trouble of all that typing, I wrote a program in Apple Business Basic to set the parameters for me. The program, PICKAFONT, sets up the Epson to one of four fonts (NORMAL, DOUBLE-WIDE, SMALL, or COMPRESSED) and to none, one, or both of these characteristics (UNDERLINED, ITALICIZED). Once the program has been executed, the Epson remains in the settings I've made.

Let's say I run the program to set the Epson to print in COMPRESSED font with UNDERLINING. Then I boot Visicalc. When it's time to print spreadsheets, they will be printed out both COMPRESSED and UNDERLINED. And using a small font like COMPRESSED (or SMALL) means that I can print more than 20 columns across on wide-sheet paper, or 12 columns across on 8.5 x 11" paper.

Now, on to print out disk catalogs and program listings. I have to reopen the printer by typing in the following:

OPEN#1,".PRINTER":OUTPUT#1 (in Applesoft: PR#1).

Then any CATALOGS or LISTS get sent to the .PRINTER instead of to the .CONSOLE.To subsequently close the printer, all I need type is

CLOSE#1 (in Applesoft: PR#0).

Good luck in running this program. I'd appreciate hearing from anyone concerning improvements to this program or comments on bettering my Basic programming techniques.

```
1000  k$=" ":x$=CHR$(34)
1010  FOR i=1 TO 24:PRINT" ":NEXT i
1020  VPOS=06:HPOS=13:PRINT"E P S O N  F O N T  S E L E C T  P R O G R A M
      "
1030  IF k$=" " GOTO 1050
1040  GOTO 1240
1050  VPOS=08:HPOS=25:PRINT"This is a utility program to select a font"
1060  HPOS=25:PRINT"for the Epson MX-100 printer to use."
1070  HPOS=25:PRINT" "
1080  HPOS=25:PRINT"Once a font is selected along with accompanying"
1090  HPOS=25:PRINT"features (underlining, italicizing) then this"
1100  HPOS=25:PRINT"program lists the features selected along with"
1110  HPOS=25:PRINT"a small paragraph.  The Epson will remain in"
1120  HPOS=25:PRINT"this setting until it is reset by software"
1130  HPOS=25:PRINT"commands, or it is powered off."
1140  HPOS=25:PRINT""
1150  HPOS=25:PRINT"I use this program to set the Epson to the print"
1160  HPOS=25:PRINT"setting I want.  Subsequent printings of disk"
1170  HPOS=25:PRINT"catalogs, program listings, or VisiCalc spread-"
1180  HPOS=25:PRINT"sheets are done in this setting."
1190  HPOS=25:PRINT" "
1200  HPOS=25:PRINT"Hit RETURN to continue"
1210  VPOS=23:HPOS=47:INPUT k$
1220  k$="Y"
1230  GOTO 1010
1240  VPOS=11:HPOS=25:PRINT"    1 -- Normal font."
1250  VPOS=12:HPOS=25:PRINT"    2 -- Double-Width font."
1260  VPOS=13:HPOS=25:PRINT"    3 -- Small font. (won't use underlining)"
1270  VPOS=14:HPOS=25:PRINT"    4 -- Compressed font."
1280  VPOS=10:HPOS=25:PRINT"WHICH FONT WOULD YOU LIKE"
1290  VPOS=10:HPOS=50:INPUT a$
1300  IF a$="1" GOTO 1350
1310  IF a$="2" GOTO 1350
1320  IF a$="3" GOTO 1350
1330  IF a$="4" GOTO 1350
1340  GOTO 1280
1350  VPOS=16:HPOS=25:PRINT"Do you want UNDERLINING     (Y or N)"
1360  VPOS=16:HPOS=48:INPUT b$
1370  IF b$="y" THEN b$="Y"
1380  IF b$="Y" THEN GOTO 1420
1390  IF b$="n" THEN b$="N"
1400  IF b$="N" THEN GOTO 1420
1410  GOTO 1350
1420  VPOS=18:HPOS=25:PRINT"Do you want ITALICIZING     (Y or N)"
1430  VPOS=18:HPOS=48:INPUT c$
1440  IF c$="y" THEN c$="Y"
1450  IF c$="Y" GOTO 1490
1460  IF c$="n" THEN c$="N"
1470  IF c$="N" GOTO 1490
1480  GOTO 1420
1490  VPOS=20:HPOS=25:PRINT"Are these choices good      (Y or N)"
1500  VPOS=20:HPOS=47:INPUT d$
1510  IF d$="y" THEN d$="Y"
1520  IF d$="Y" THEN GOTO 1540
1530  GOTO 1240
1540  VPOS=09
1550  HPOS=20:PRINT"Now the program will conclude by sending a small
1560  HPOS=20:PRINT"paragraph to the Epson to be printed.  If you now
1570  HPOS=20:PRINT"want to print out disk catalogs or list this program,
1580  HPOS=20:PRINT"let's say, you'll need to type in the following to
1590  HPOS=20:PRINT"reopen the printer --
1600  HPOS=20:PRINT"
1610  HPOS=20:PRINT"OPEN#1,"X$".PRINTER"X$":OUTPUT#1
1620  HPOS=20:PRINT"
1630  HPOS=20:PRINT"Then, when you type LIST or CATALOG, the LIST or
1640  HPOS=20:PRINT"CATALOG gets sent to the .PRINTER and not to the
1650  HPOS=20:PRINT".CONSOLE.  To close the printer, you would type
1660  HPOS=20:PRINT"
1670  HPOS=20:PRINT"CLOSE#1
1680  OPEN#1,".printer":OUTPUT#1
1690  REM Reset the software codes
1700  PRINT CHR$(27)"@"
1710  IF a$="1" THEN GOSUB 1880
1720  IF a$="2" THEN GOSUB 1900
1730  IF a$="3" THEN GOSUB 1930
1740  IF a$="4" THEN GOSUB 1970
1750  IF b$="Y" THEN GOSUB 2010
1760  IF c$="Y" THEN GOSUB 2080
1770  IF a$="1" THEN PRINT"Normal font in use."
1780  IF a$="2" THEN PRINT"Double-Width font in use."
1790  IF a$="3" THEN PRINT"Small font in use."
1800  IF a$="4" THEN PRINT"Compressed font in use."
1810  IF b$="Y" THEN PRINT"    Font will be underlined."
1820  IF c$="Y" THEN PRINT"    Font will be italicized."
1830  PRINT:PRINT
1840  PRINT"    Four score and seven years ago out fathers brought forth"
1850  PRINT"on this continent, a new nation, conceived in Liberty, and"
1860  PRINT"dedicated to the proposition that all men are created equal."
1870  GOTO 2110
1880  REM Set NORMAL font on
1890  RETURN
1900  REM Set DOUBLE-WIDTH font on
1910  PRINT CHR$(27)"W"CHR$(1)
1920  RETURN
1930  REM Set SMALL font on
1940  PRINT CHR$(15)" ";CHR$(14)" ";CHR$(20);CHR$(27);CHR$(83);CHR$(0)
1950  PRINT CHR$(27)"A"CHR$(6):REM Set horizontal spacing at 6/72" per line.
1960  RETURN
1970  REM Set COMPRESSED font on
1980  PRINT CHR$(15);" "
1990  PRINT CHR$(27)"A"CHR$(9):REM Set horizontal spacing at 9/72" per line.
2000  RETURN
2010  REM Set UNDERLINING on
2020  REM Ignore UNDERLINING if SMALL font was selected.
2030  IF a$="3" THEN b$="N"
2040  IF a$="3" THEN GOTO 2070
2050  PRINT CHR$(20);CHR$(27);CHR$(52)
2060  PRINT CHR$(27);CHR$(45);CHR$(1)
2070  RETURN
2080  REM Set ITALICS on
2090  PRINT CHR$(27);CHR$(52)
2100  RETURN
2110  END
2120  REM This program was written December 12, 1982 by Jim Linhart
2130  REM of Berkeley, California and is donated to the Original
2140  REM Apple ///'ers Users Group.  It is in the public domain.
```

Normal font in use.

Four score and seven years ago out fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Normal font in use.
Font will be underlined.
Font will be italicized.

Four score and seven years ago out fathers brougl
on this continent, a new nation, conceived in Libert
dedicated to the proposition that all men are create

Normal font in use.
Font will be italicized.

Four score and seven years ago out fathers brougl
on this continent, a new nation, conceived in Libert
dedicated to the proposition that all men are create

Normal font in use.
Font will be underlined.
Font will be italicized.

Four score and seven years ago out fathers broug
on this continent, a new nation, conceived in Libert
dedicated to the proposition that all men are create

Double-Width font in use.

Four score and seven y
on this continent, a new n
dedicated to the propositi

Double-Width font in use.
Font will be underline

Four score and seven y
on this continent, a new n
dedicated to the propositi

Double-Width font in use.
Font will be italicized.

Four score and seven years ago out fathe.
on this continent, a new nation, conceived i:
dedicated to the proposition that all men ar

Small font in use.
Font will be italicized.

Four score and seven years ago out fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Small font in use.

Four score and seven years ago out fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Compressed font in use.

Four score and seven years ago out fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Compressed font in use.
Font will be underlined.

Four score and seven years ago out fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Compressed font in use.
Font will be italicized.

Four score and seven years ago out fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Compressed font in use.
Font will be underlined.
Font will be italicized.

Four score and seven years ago out fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Compressed font in use.
Font will be italicized.

Four score and seven years ago out fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

CHILDREN'S BENEFIT COMPUTER FAIR
by Allan B. Woods

The Childrens Benefit Computer Fair will be held on April 9 & 10 at Del Valle High School public events center, 1963 Tice Valley Boulevard, Walnut Creek, CA.

The fair will be open Saturday, April 9 from 9:00 A.M. until 6:00 P.M., and on Sunday, April 10 from 10:00 A.M. until 5:00 P.M.

Ticket prices are:
Adults - $50.. per day advance purchase
Children under 12 w/adult $2.00
Under 5 years - free!

Tickets and information can be obtained from the Contra Costa Children's Council, 2253 Concord Boulevard, Concord, CA 94520 - (415) 676-5442. Please include self-addressed stamped envelope when ordering.

The Children's Benefit Computer Fair is a fund raising effort being sponsored by a consortium of non-profit child care agencies serving the greater Concord area.

CHILDREN'S BENEFIT
COMPUTER FAIR
2253 Concord Blvd., Concord, CA 94520 (415) 676-5442

Notes from the International Core

This months DOM was put together by the IAC software chairman, Dr. Smith, and contains "Basicalc". This is the program that was listed in "Softalk" Magazine sometime ago and is written in Applesoft. Included in the disk are many application templates for the program. On occasions the IAC will be sending a second disk which will contain programs not in DOS. These could be CP/M, Forth, Fortran, Pascal, or other operating systems/languages. The first of the special disks will be next month and contain CP/M Utilities.

The first Update for your "Apple Tech Notes" is now back from the printer and has been mailed. U.S. clubs should be receiving it now and outside of the U.S. within 6-8 weeks.

The IAC now gives a discount to members of IAC clubs for the "Apple Orchard". In this mailing are subscription forms to be used requesting the **discount.** The new 9 issue price is $19.50 with a $2.50 discount - $17.00 for 9 issues.

COMING SOON

The A.U.G.E. (Apple User Group Europe) has donated a data base management system for IAC clubs. It is called IACBASE and is in Pascal. As soon as more documentation is received this will be sent to each member club.

JOB WANTED

Looking for jr Prog/SA Position

Some exp writing file specs and code for MicroData REALITY sys.

5 yrs bus/accting bckgd.

2 yrs UC Extension DP courses.

If you know of openings please call Bob Hosek, (415) 525-6553
1240 Milvia St, Berkeley 94709

INTERNATIONAL APPLE CORE ™

APPLE ORCHARD SUBSCRIPTIONS

For IAC Club Members Only

908 GEORGE STREET, SANTA CLARA, CALIFORNIA 95050

☐ 9 issues for only $19.50
($29.25 at newsstands)
$17.00 for members of IAC clubs only

☐ Foreign: 9 issues
$36.00 Surface
$29.00 for members of IAC clubs only

☐ Payment enclosed
(US funds only)

Start my subscription with issue # _____

Name _____

Address _____

City _____ State _____ Zip _____

☐ Name of Club _____ Signature of club officer _____
Valid only with signature

InterBank # ☐☐☐☐ 4 digit number above your name on MasterCard™ ☐ VISA

Charge Card # ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Credit Card Signature _____ Card Expires

The San Francisco Apple Core Presents
The Second Semi-Annual Apple Swap Meet

Saturday, February 19, 1983

Fort Mason National Recreation Center
Building C., San Francisco, 10 AM to 2 PM

Used Apple equipment (hardware and software) will be bought, sold, and swapped; and at 1 PM, an Auction will be held.

You are advised to arrive early, while material is still plentiful

---

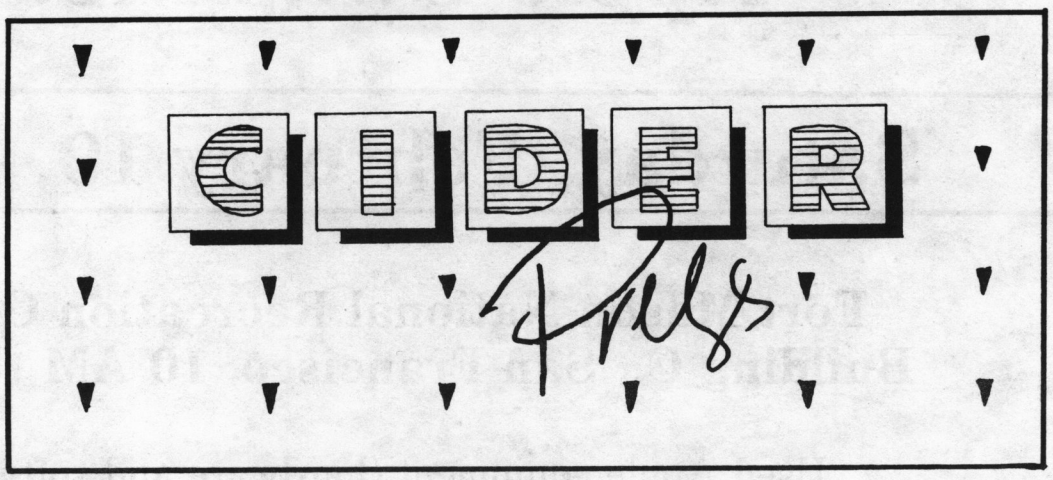
Admission: Free to all San Francisco Apple Core members (bring and show your membership card). Non-members, $1.00 at the door.

Booth space: There will be a limited number of booths available at this swap meet. For details, contact Ken Silverman at (408) 727-7652, between 9 AM and 5 PM. First come, first served.

COVER CONTEST WINNERS ANNOUNCED
by Gene Wilson

The "Cider Press Cover Contest" has officially closed, and the winners have been chosen (by yours truly). The "Cider Press", as benefactor, has sponsered this contest in the hopes of improving the quality of the entire publication. The winning entry does precisely that.

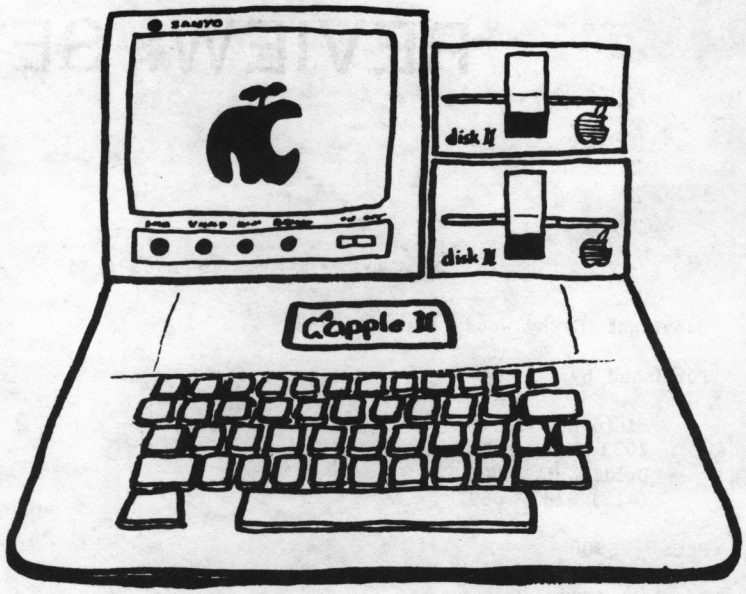
The winner of our contest is Sean Adams, of San Rafael, California.



The winner is also Lora Asdorian, of Bedford, Texas.

And the winner is Jim Linhart, of Berkeley, California.

Cider Press



That's right, there is a three way tie for the first prize, which is an 'Apple Core Library' (or equivalent merchandise from our sales table). All three of these winners have worked to put their ideas into an enjoyable form for the rest of the membership. It is our hope that future contests will be as successful. Our thanks for a job well done.

C/M'83
A FEW NOTES
by Gene Wilson

The scene of this exhibit was the new Moscone Center, in San Francisco. Held from January 21 thru 23rd, this event looked to be a 'massive' success. All manner of hardware and software was in evidence; some of the displays were simply mobbed by the curious crowd.

An example of the 'new' technology on display for the Apple][was "the Third Alternative" by Saybrook, which consists of a 68000 co-processor card. The hardware will include (available soon?) a Motorola 68000 32-bit microprocessor, 8MHz, 128k RAM (expandable to 512k), etc. Software will include the UCSD p-System (as its standard operating system), BASIC, FORTRAN-77, and Pascal compilers. Also on the slate for future release is an Applesoft-compatible 68000 BASIC, CP/M-68K, and UNIDOS (UNIX Version compatible). The literature from Saybrook warns that the card 1.) generates heat (caution to use fan), and 2.) draws power (advises the use of an 'improved' power supply). Further information is available from:

Analytical Engines, Inc.
3415 Greystone
Austin, TX 78731
(512) 346-8430

On the 'software' front were demonstrations of word processing packages such as "Spellbinder", database systems such as the "Condor Relational Database System, and a show favorite, "dBase][".

Among the distributors, there was a 'healthy' competition, and a number of show 'deals' were offered. The 'deal' which was impossible to pass up was put forward by MICRO-AP. "SELECTOR V Database Management System" was offered at half price ($450 vs. the normal $900). If this wasn't convincing, a Show Special 'Evaluation Plan' was offered to end users to try the product for two weeks for the sum of $45. (The only catch here is that until a special code is given to the program, only sixty records can be placed on any single file. This limit is enough to see if the product works as advertised. See this month's review by Woody Liswood). For further information contact:

MICRO-AP
7033 Village Parkway
Suite 206
Dublin, CA 94566
(415) 828-6697

Among the many word processing packages was "SPELLBINDER". This program has some very nice features, including a form of 'proportional spacing' for incremental printers (each character has a pre-defined width stored in a table), and two column printing is also supported by one of the many 'macro' enhancements. SPELLBINDER is a product of:

LEXISOFT INC.
Box 1378
Davis, CA 95617
(916) 758-3630

REVIEW: SELECTOR V

SELECTOR

Copyright (C) by Woody Liswood 1983

Published by:

Micro-Ap, Inc.
7033 Village Parkway, Suite 206
Dublin, CA 94568
(415) 828 - 6697

Price: $900

System: CP/M

SELECTOR is what you and I would call a data base. They (the publisher) say that it is an "Advanced Applications Management" system. What I think is that it is the best Data Management System which I have found so far which runs in the Micro computer environment. Notice I said "Micro" not APPLE. That is because when you are dealing with CP/M, you really are working with programs that run on a multitude of machines, not just a single one. Today it seems that all programs must first run on the IBM PC, then in APPLE DOS or CP/M worlds.

SELECTOR is a relational application tool. It seems to do more that the "typical" Data Base Management System which I have seen, so I will go along with the tool bit. When talking about **APPLE** computers, it makes sense to compare **SELECTOR** to **DB MASTER** or to **DBASE II. DBASE II** seems to be the best selling data management system in the CP/M world. I still think that **DB MASTER** is the best of the Data Base systems which run in **APPLE DOS.** I also now feel that **SELECTOR** is the best of the systems which I know about in the CP/M world also. You probably want to know, at this point, why I didn't pick **DBASE II** as the best CP/M program. Two reasons. First, **DBASE** is a language, not a menu driven system. **SELECTOR** is a menu driven system. Second, **SELECTOR** can do essentially everything you can do with **DBASE** and it seems to run faster. Micro-Ap says it sorts at least 9 times faster. I don't particularly like comparative reviews so I will not try to match feature for feature with any other CP/M system. I will let you interpret my facts and opinions as you see fit.

DETAILS

Rather list the technical details out all at once, let me take you through the main menu mention the details as you would use them. Otherwise this review would sound like sales literature.

The first thing to consider is that **SELECTOR** can keep six data files open at a single time. **DBASE** is limited to 2 and **DB MASTER** is limited to one. A data file is limited to the data which you can display on one 80 column by 24 line screen. There is also a limit of 89 fields per record. **DBASE** only has a 32 field limit and 100 fields are the limit for **DB MASTER.** **SELECTOR** has a limit of 1694 characters in a single record. they calculate that this way: "On a screen with 24 rows, you may use 22 rows times 79 characters per row, or a total of 1,738 characters available for the entire Data Entry Screen. That includes field names, the brackets [] which must surround a field name, the length of the data field, and any comments or graphics you might want to use. If you do not assign any field names to any of the fields, you can have a maximum record length of 22 rows times 77 characters, or 1,694 characters."

INSTALLATION

SELECTOR has the typical install program found in most good CP/M software. These install routines are there because of the variety of techniques which the various hardware manufacturers use to control how the cursor is placed on the screen. You run install and are given your choice of a variety of terminals which you can say you have. In addition, if you have a non-standard or probably a new terminal not yet on the list, you can also supply the cursor codes directly through a prompting process and still get **SELECTOR** running on your system.

The program has a full screen editing capability for the design of data entry screens. It comes supplied with the standard WordStar defaults for cursor and screen movements, along with some control sequences used specifically for the data base. And, there is a goodie here. You can customize the cursor movements any way you wish. There is a screen with all of the cursor control movements listed along with the control sequence needed to cause those things to happen. If you want your cursor left command to be a Control-J rather than a Control-S, you say that you would like to change the cursor left command, and the program then requests that you press the sequence you wish. It reads what you press and installs that into the program. This will now be shown on all of the help screens as well. Notice I did not have to look up the **HEX** or **DECIMAL** equivalents of any of this.

On my **APPLE,** I run a **VIDEX** Keyboard Enhancer as well as their 15 key function strip as an integral part of my system. So, when I sat down to do this, I discovered that the **VIDEX** Function strip just generates a standard ASCII code not a special set of codes. **SELECTOR** read the first function key as a standard 0 which, if I had used it, would have meant that I could not enter any numbers into the

system. The way around it was to load the keyboard encoder with the same WordStar control sequences which were already contained within **SELECTOR** then **SELECTOR** read those keys as if they were standard function keys on the typical CP/M based system. I also installed **SELECTOR** on my **ESCORT** portable computer which has a set of 10 function keys. Those keys generated a lead-in character when pressed so they were considered unique to the program. For example, the F1 key, when pressed, was read by **SELECTOR** as a Control-A A, the F2 was a Control-A B. **SELECTOR** then read those as the new defaults and I can use the function keys to control the data entry screen.

After you accept the screen defaults you are prompted with a series of questions related to the other system defaults. You are asked how many drives are going to be connected to the system. My **APPLE**, with two floppy's and one hard disk responded as ABCD. Then you are asked the micro-processor speed in Mhz. It defaults to 4. This will determine some how long certain messages remain on the screen when you use the program. You are next queried as to printer width and page length. Although these are defaults, you can over-ride them in the report generators so it is not too critical to be absolutely accurate here.

Next is the number of fields you want for any record. The default is 36. If you are going to use the maximum of 89 fields, you must enter that number here. You cannot over-ride this number during the program operation. If you are going to do a lot of smaller applications, don't just automatically pick the large number. You can always rerun the installation if you need the 89 fields. If you go with 89, you will always be wasting space in memory and it may significantly slow the program down when it is not needed. On-the-other-hand, the number you select here for a field number limit also sets a limit on the number of procedure steps you may have available as part of a single procedure. These procedure steps, as you will read later, are perhaps the single most valuable feature of **SELECTOR**.

KEY RESERVE FACTOR

Next is a "KEY RESERVE" factor. It defaults to 0. Advice is to leave it alone. What it does is determine the amount of array space the program sets up in memory for operation. This also lets the program run in a variety of systems which may, after CP/M gets installed, have varying amounts of memory left for the program. In fact, I tested the system with a 20 field file which contained over 500 records and with the number of fields set to 89. I started having memory error problems around 400 records. A call to Micro-Ap suggested that I change the Key Reserve to 2 or more. I did, the problem when away. Then, I discovered in the manual a paragraph which said that if you were having memory problems to increase the key reserve number. So I really didn't have to call after all.

Next you are asked if you would like to eliminate the low level help messages. And, lastly, you are asked if you would like to name a special program which you can call from the program. For example, you may wish to always exit **SELECTOR** by running **STAT** or **CAT**.

After this, you are asked to enter your name, address, and telephone number. These become system defaults which you can use in your printed reports. They are also an integral part of the "protection" scheme.

Here's what happens. When you first receive **SELECTOR** you have a system which can only address 60 records. That way you can work with the system and make certain that you wish to purchase it. However, to get that 60 record limit released, you must call Micro-Ap and tell them the serial number of your copy of the system and then the exact wording you placed in the name and address lines. The Micro-Ap folks then give you a special code to key in which releases the system to its full potential.

DOCUMENTATION

I had a interesting relationship with the documentation. It is a bound 326 page book. The book was typeset using **SPELLBINDER**, a PS 10 type wheel on a daisy wheel printer, and is well organized and easy to read. Here's what happened to me. When I first read the manual, I thought that it was the greatest thing since sliced bread. It appeared to be clear, concise, and to describe the program exactly. Then, when I booted the program and actually started to use the program, the manual began to make less and less sense. Then, after using the program for a few days, the manual again made lots of sense.

The problem, to me, is that the manual is organized backwards. If you purchase **SELECTOR** you should read the first two chapters first. These give you some information about the program and walk you through the install program. Then, skip to chapter 6. That chapter has sections A through M and tell you how to create a data entry screen, define your fields, enter and recall data, do selects and sorts, create columnar and formatted reports as well as other goodies. In fact, that chapter addresses all of the functions that are mentioned in the main menu. After you read that section, go back and read chapter 3, which is a tutorial; chapter 4 which explains the various types of fields; and chapter 5 which explains the boolean operators and connectors as well as procedure steps.

Pay a lot of careful attention to the chapter on procedure steps. To me, these procedure steps represent a creative way to have both a programming language and a menu driven program. They are for me the feature which places **SELECTOR** a cut above the other programs I have used.

Then, and only then, should your run the demo which comes with the program. The problem I have with the demo and with the tutorial is that they use a very sophisticated and complicated project as the demo. When you still don't know how to run the system, the demo and tutorial are so complicated as to have very little meaning to the first time user. In all fairness, I should say that after I learned the system, I did look at the demo and the tutorial so that I could see how the author did certain rather complex functions. I think that there should be at least two demo programs. One with this complex system, and one for the beginner, which go through the basics of a simple system.

DEMO

SELECTOR has a very interesting function which they use for the demo but which you, as a system designer, will find an invaluable feature. You can boot **SELECTOR** three different ways. First is just call up the program. Second you would call up the program with a space and the words REC. i.e. **SELECTOR REC.** Or you can call it with the extension PLAY. (**SELECTOR PLAY**) When you go regular, the program operates as you would expect. When

you boot with **REC**, a file is created which captures everything which appears on the screen, all of your keystrokes, and all of the data which you might be entering. You can then use a word processor and clean up that file to contain only the commands and responses you wish. Then, you can boot with the **PLAY** extension and **SELECTOR** will treat the file you created with the **REC** mode as a command or EXEC file and run itself.

That means you can design a system, make certain it works, then set up a command file which will eliminate the need for operator intervention. You would let the operator do data entry into a holding file. Then you would have them run the play mode and call up the command file. Then, **SELECTOR** would update up to 5 other files on a single pass through the data file and run any number of procedures you wish, and print any of the reports you desired. Yes you can have the machine run all day with just one command.

DOES THAT SOUND LIKE MAIN-FRAME CAPABILITY? I think so. The difference is speed and storage, not capability.

Back to the Demo. **SELECTOR** comes with a demo which sets up a multi file inventory system which will show you how to use a multi file system and take advantage of all of **SELECTOR**'s procedures. After you have read the manual, in my suggested order of course, you might take a few hours to run the demo and watch what the system does. In all honesty, I found the demo less than useful until I had worked with the program for a few days and understood what it was trying to show me.

MAIN MENU

When you boot the program, you are asked for the current date and for your name. These are used as part of an automatic audit trail. Then the main menu appears. At the top you are informed as to the version number, your serial number, whether you have an unlimited or 60 record system, and whether you have the record or playback mode on.

Then, you are given 14 choices. Many of the choices are in pairs. The concept is to create something with one part of the menu, and do something with the other part.

(A) Screen/Record Definition
(B) Data Entry/Recall/Edit
(C) Select/Sort Definition
(D) Select/Sort/Execution
(E) Label/Column Report Def.
(F) Label/Column Report Exec.
(G) Formatted Report Def.
(H) Formatted Report Exec.
(I) File Translation
(J) Rebuild Index Key File
(K) File Conversion
(L) Batch Post to File(s)
(M) File Query
(X) EXIT to System
(N) X:XXXXXXXX.COM

Before starting on each of the menu items, there are a few features about **SELECTOR** which are consistent with all of the menu features. Each of the menu selections from the main menu bring you to a sub-menu with a series of additional choices. Each of those sub-menu's will allow you to do a normal exit back to CP/M, to run the next logical part of the program, to create or print or re-design whatever the menu function was all about. I have a small criticism here. In each of these sub-menu areas, you are asked to give the drive designation and file name of the file you wish to re-design or use. However, there is no provision for displaying the disk directory to see what is really on there. In a program this sophisticated, that seems like a necessity. In fact, if I were designing that particular interface, I would have a special set of file extensions which are generated automatically by each of the sub-menu's. Perhaps ".BCH" for the Batch Post to File file, ".FMT" for a formatted report definition file and ".CLM" for all columnar report definition. Then, when the user wanted to use a previously developed file, I would have the program display the appropriate files as designated with the appropriate file extensions. That way you could select immediately from the set of possible choices. I would limit the files displayed to only the extensions. Most CP/M folks seem to use hard disks. Without that limit, there could literally be hundreds of possible file names shown and the user would still have a difficult time finding the correct one. And, if I were a really sophisticated programmer, I would assign a number to each of those files so that the user could call the file into use by selecting a number rather than keying in the entire name. "Are you reading this Micro-Ap?"

SCREEN / RECORD DEFINITION

This is the first menu item and one of the most important. Using this module allows you to create the display screen you wish and to define the field types that are contained in the screen. When you select **A**, you get a new menu which allows you to create or re-design the screens, to create or re- design the data format, to print a set of file definitions, to exit the system and to go direct to the data entry module.

In **SELECTOR**, you create your data schema by first designing the screen. You do this with full screen editing. This means that you are shown a blank screen with a live cursor. You can move the cursor anywhere you want and do anything you want. You designate a field name by enclosing it in square or round brackets. [Name] and (Name) are both valid. You define how long you want the field to by by following it with underlines or with asterisks. [Name] _____ or (Name) ********** are both valid. If you need a decimal number, you just include a decimal point in the display. [Cost] _____.__ is correct. There are commands to center lines, move lines, insert or delete lines, and to draw boxes around material. When you are finished, **SELECTOR** saves the file on disk with a SCR extension.

SELECTOR defines the field types by displaying the screen to you, numbering the fields, and allowing you to define the field types one at a time. The field order goes from left to right and from top to bottom. So, when you design a form, you should pay attention to the way in which you want the operator to enter the data, since the fields will **ALWAYS** be in that order.

This would be one of minor criticisms of **SELECTOR.** I would like to be able to design the screen, then be able to specify the order in which I wished the cursor to move from field to field during data entry or recall. Oh well, everything can not be perfect.

Screen Format for B:HRPERSNL.SCR as of JAN 28, '83.

```
                                PERSONAL DATA

[ID #] _____  [Phone #] ____      [Hours Worked] __.__  [Status] __

[Last Name] _____________ [First Name] _____________ [Nick Name] _________

[Social Security Number] ___-__-____  [Date of Birth] ______
    [Address] ____________________________
    [City] _________________ [State] __  [ZIP] _____
[Home Phone] ____________  [Emergency Name] ________________________
                           [Emergency Phone] ___________
                           [Emergency Phone] ___________

[Married] _ [Spouse Name] _______________ [Spouse DOB] ______
[Hire Source] ___  [Hire Date] ______  [Service Date] ______
[Sex] _  [EEO] __ [Ethnic] __ [Handicap] __  [Draft Status] __ [Reserve] __
[Viet Veteran] _ [USA Citizenship] _ [Visa Type] __  [Country] ___
[Educational Level] ___  [Major] ___ [Year Graduated] __
[School] ___  [Year B.S.] __ [Language #1] ___ [Language #2] ___
[Security Clearance] ___
[Termination Reason] ___  [Termination Date] ______
[Remarks] ____________________________________________________________
```

The field types available are Alphabetic, Numeric, or Dates. You can then specify whether the field is to be right justified, is to be a must enter field, or a skip entry field (for when you want a procedure step to calculate a value and place it into this field, or batch post it into the field, or call a record off another data file and place it into the field, or...). Then you specify whether you want the field to be a non-unique or a unique key. You may have every field a key field if you wish. If you need it, you may also have a series of separate field keys which are concatenated. That means you can create a separate key which is the combination of two or more fields. (Concatenate seems to be one of the favorite words of the authors of the manual for **SELECTOR**. Just thought you might like to know that.) And, I discovered, you are not locked into your initial choices. After you have data in the system, you may decide that you wish to recall data on a field which you had not made a key. You would just redefine the data definition to make that field a key field, then run the rebuild index key module in the main menu and that field is now a key.

If you defined a date field, you must say whether you want the date displayed as mmddyy, ddmmyy, or yymmdd. You do have a choice.

You may also set data limits to a field. You may say to reject all entries between 50 and 500. And, you can if you wish verify the data in the field by displaying the contents of another field in a separate file. For example, you might be entering job code numbers. Now, we all know that you don't memorize pages of numbers. So when you are entering the number, **SELECTOR** will look up that number in a job code file, and display to you the title so that you can be certain that you really did enter the job code correctly. This feature is a very valuable one but I think that it is only half implemented. What would also be valuable or perhaps even more valuable would be that after the data is displayed, I would like to be able to have that data entered directly into the data base as a entry.

You can get that automatic entry now, but it takes two steps. Here's the same problem. You verify the job title when you enter the job code. The next field says "TITLE" but you call it a skip field. Then, you do a batch post which reads the job code field, goes to the title file, and inserts the title back into your master file. On the other hand, maybe you don't want to save the title in your master file and in your title file because you can always print the title in a report by using the job code from the master to look up and use the title from the title file. Now you may see why I have switched all my data base applications to **SELECTOR**. I have a choice. If I want the title in the master file, then I can get it their without keying it. If I only want to print the title in certain reports, I can save data space by carrying it in only one file and coding it in the master. How about that?

When you are done with the screen and record definitions, you return to a sub-menu which allows you to return directly to the main menu, go directly to the data entry and recall sub-menu, exit back to CP/M, or print the screen and record definitions for documentation and future use. You should always print those immediately. **SELECTOR** uses file numbers and field numbers in its reporting and file manipulation modules. You need to know what number you assigned to each of the fields. Following is a sample Screen and Field definition for a **HRIS** which I have designed to run in the **SELECTOR** environment.

DATA ENTRY AND RECALL

The most critical boundary between you and the machine is the data entry and recall features of the program. If you have been working with any of the data management programs in the DOS world, you have a good feeling for just how slow the **APPLE** can work. The response for **SELECTOR** is (or seems to be) a great deal faster than I have found for other data management programs.

When you get to the Data Entry and Recall sub-menu, you can add new records, view and edit existing records, change to other data files, return to the main menu, or exit the system.

B:HRPERSNL.DFN Record Definition Data Drive: B 0 Records as of JAN 28, '83.

Fields: 44 Record length: 321 Key record length: 16

FIELD NUMBER / NAME TYPE	FIELD LENGTH	KEY LENGTH	DATE FORM	VERIFY CALLED VERIFY FILENAME / KEY / FIELD#	MIN / MAX VALUE
1 / [ID #] NU	5	5			
2 / [Phone #] N	4	0			
3 / [Hours Worked] N	5	0			
4 / [Status] N	2	0			
5 / [Last Name] AK	13	7			
6 / [First Name] AK	13	5			
7 / [Nick Name] AK	9	5			
8 / [Social Security Number] N	3	0			
9 / [Date of Birth] D	6	0	MDY		
10 / [Address] A	30	0			
11 / [City] A	17	0			
12 / [State] A	2	0		B:HRSTATE./1/2	
13 / [ZIP] A	5	0			
14 / [Home Phone] N	12	0			
15 / [Emergency Name] A	24	0			
16 / [Emergency Phone] N	11	0			
17 / [Emergency Phone] N	11	0			
18 / [Married] A	1	0			
19 / [Spouse Name] A	15	0			
20 / [Spouse DOB] D	6	0	MDY		
21 / [Hire Source] N	3	0		B:HRHIRSOR./1/2	

DATA ENTRY

Data Entry is accomplished by moving from field to field and keying in your data. When you get to the bottom of the screen, the cursor automatically moves to field number one. You can page through your fields with the cursor and make corrections and change data as necessary. When you are happy with the results, you key a Control-W to write the record to the disk. The cursor then returns to field number one. Now for something interesting.

The data from the record you just wrote to the disk stays visible and actually becomes default values in all the fields. You now may only change one thing and write the entire new record to the disk. **SELECTOR** gives you four methods to do this. First, you carriage return to the field you wish to change and use the arrow keys to position the cursor over the item you want to be changed, change that character, carriage return to a new field, then do a Control-W to write the new record to the disk. Or, two, you can get to the field, clear that field then key in all new data. Or, three, you can have as a default that the fields will clear automatically when the cursor gets to them. Or, four you can clear all of the fields of data before you start to enter the next record. You can choose the system which best fits your style of operation.

SELECTOR also does some checking for you as you go along. If you have declared a unique key field, **SELECTOR** will not allow you to enter a record with the same key. If you have non-unique keys, **SELECTOR** will not bother with any message. But, if you forgot whether or not you just entered a record, if you had, **SELECTOR** will notify you that this is a duplicate record and ask your permission to add it to the file.

DATA RECALL

Having worked with **DB MASTER** over the last few years, I know how slow a program can work. **SELECTOR** will amaze you. You browse through your records almost as fast as you can press the carriage return. You can browse on any key field you selected. When you get to a record you want to work on, you press an **E** for edit, the cursor then jumps to the first field on the screen, then you move through the record doing what you want to do. When you are done editing, a Control-W writes the record to disk. You select the search field by typing only enough of the field name for it to be unique. **SELECTOR** then moves the cursor to that field and clears it waiting for you to key in the appropriate data. After you have displayed the data a carriage return moves you to the next highest record, a **L** moves to you the next lowest record, a **P** will print that record off to the printer, a **E** goes to the edit mode and a **D** will delete the record. When you have browsed to the end of the file, **SELECTOR** will notify you that there are no more records in that file.

SELECT DEFINITIONS

Before exploring the two major reporting options, let's look at the way in which **SELECTOR** allows you to select records for a report. **SELECTOR**, as will most programs, always let you work with all the records in your file. And, **SELECTOR** will, during its normal report procedure allow you to change the key file, the pointer file, and/or the range of records you wish to print on that particular pass through the data file. However, that is not always enough. What if you wanted a list of left handed blue widget makers who lived in ZIP codes 94213 and 94127, who were currently earning over $100 per year. Yes, you do have that information in your master file.

You turn to the select/sort definition module from the main menu. You then wind up at a table. You pick a line and selector will lead you through a set of decisions. First you must say what file you wish to use, then what field and test you wish to use. **SELECTOR** has full boolean conditional tests. You can compare with values, other fields, computed fields, or anything you want. If you are comparing alphanumerics you must be careful about upper and lower case letters. Woody is not equal to WOODY. If you compare dates, you must enter them in year-month-day order. As you build this table, you enter comparisons on odd numbered lines and connectors on even numbered lines. That means that or's and and's go on even numbered lines. When done, you key a & to exit from the entry routine then name the definition and notify **SELECTOR** what disk you wish it to be stored on. Then, if you wish, you create that list.

However, **SELECTOR** does not create a separate data file for you with just that set of data. What it does is create a separate index or pointer file which is run against your main data file. The report generators then use that pointer file to select the records you wish to use. By-the-way, if you wish to use that pointer list to create a new data file with just that information, you can also do it that way. Again, the choice with this program is all yours.

SORT DEFINITION

Although you can have each of the fields as a key field, that would begin to take a lot of room on your disk. So, there may be cases when you would rather do a report, a batch post, or a file conversion by some order other than using a key field or with descending or ascending sorts within the same field. I think on this one, a paragraph from the documentation says it best:

"...A Sort Definition allows you to define the order in which you wish the records to be read for the purposes of the Report, Batch Post or File Conversion. You can sort on up to 16 fields in any combination of independently ascending or descending orders that you wish. For example, you might want to print a report of transactions in order by ascending customer numbers, but you might want to see the most recent transactions listed first, so within the customer number in ascending order, you would request the date fields to be sorted in descending order."

It works by asking you to name the major sort first, whether or not it is an ascending or descending sort, and what the field number is. You did remember, by now, to have your printed screen and field definition list with you. As in all the modules, you may execute or print the definition as part of the sub menu's.

COLUMNAR REPORTS

SELECTOR has two different reporting formats. The first is the columnar report section. Using this feature you can create the typical columnar report with a multitude of page breaks, and sub and grand totals. One of my few criticisms of **SELECTOR** comes here. You define the report by saying which fields and field numbers you wish. **SELECTOR** defines the column width by measuring either the field name or the data size, which ever is greater. That means if you have a field labeled Salary Grade and that field is a two digit numeric, the columnar report generator will print Salary Grade at the top of the column and take up 12 spaces rather than two. You may not over-ride that feature. You can, create a columnar report using the formatted report module so perhaps this is not too bad of a criticism. However, you may have some interesting work to do to get the formatted report module to do page breaks with sub-totals and grand totals the way you want them. (more about that in a few paragraphs)

Another small criticism I have with **SELECTOR** also is in this section. When you ask for a page break when a specific field changes, you do not get a page break when there is only one record for that page. That, to me, is a bug. However, the designers of the program told me that it was a specific design feature. I said that if I had wanted more than one set of numbers on a page I would not have asked for a page break when the field changed. They said that it did not make sense to waste a whole page of paper for one record. I said that when the only thing that happened was a dotted line separating that record from the next set, the program did not print out sub-totals as I wished. They said that you did not have a sub-total with only one record. They said that they would take my comments under advisement and perhaps change it at some point in the future. I said good....send me the update when you do. So far, I am not going to hold my breath. I suppose that is why programmers program and users use. At least the **SELECTOR** folks are willing to listen to their customers and really seem to care what their users think about the program.

PROCEDURE STEPS

This is a good point to talk about procedure steps. This for me was the main reason to cause me to switch all my data file operations from **DB MASTER** to **SELECTOR** (as well as the speed in which it operates).

You may have as many procedure steps as you had as field defaults during your installation . For example, if you said that your system default was 89 fields, then you could have 89 procedure steps for one single procedure. If you said that your system would only work with 32 fields per record, then you would only be allowed 32 procedure steps for a separate procedure.

The first thing which you do in any of these modules is to set up a file calling sequence. This is true for procedures, as will as for all of the reporting modules. Since **SELECTOR** can have up to six data files open and active at the same time, you need to tell the program just what six fields you wish to use. These then receive a number and you may refer to them during the creation of your goodies.

A procedure is a set of steps of things which you want to do every time a record is processed.

B:AEROLEVL.BCH BATCH POST TO FILE(S) MODULE as of JAN 28, '83

Files Used
1-AEROJOBS

BATCH POST TO FILE(S) MODULE Processing Procedures

```
 1 If [Total Points] of (AEROJOBS) LE 75  then . . .
   GOTO 32
 2 If [Total Points] of (AEROJOBS) LE 86  then . . .
   GOTO 34
 3 If [Total Points] of (AEROJOBS) LE 103  then . . .
   GOTO 36
 4 If [Total Points] of (AEROJOBS) LE 118  then . . .
   GOTO 38
 5 If [Total Points] of (AEROJOBS) LE 135  then . . .
   GOTO 40
 6 If [Total Points] of (AEROJOBS) LE 160  then . . .
   GOTO 42
 7 If [Total Points] of (AEROJOBS) LE 185  then . . .
   GOTO 44
 8 If [Total Points] of (AEROJOBS) LE 216  then . . .
   GOTO 46
 9 If [Total Points] of (AEROJOBS) LE 253  then . . .
   GOTO 48
10 If [Total Points] of (AEROJOBS) LE 294  then . . .
   GOTO 50
11 If [Total Points] of (AEROJOBS) LE 341  then . . .
   GOTO 52
12 If [Total Points] of (AEROJOBS) LE 396  then . . .
   GOTO 54
13 If [Total Points] of (AEROJOBS) LE 463  then . . .
   GOTO 56
14 If [Total Points] of (AEROJOBS) LE 538  then . . .
   GOTO 58
15 If [Total Points] of (AEROJOBS) LE 627  then . . .
   GOTO 60
```

```
76
   [Evaluation Level] of AEROJOBS=26
77
   GOTO 89
78
   [Evaluation Level] of AEROJOBS=27
79
   GOTO 89
80
   [Evaluation Level] of AEROJOBS=28
81
   GOTO 89
82
   [Evaluation Level] of AEROJOBS=29
83
   GOTO 89
84
   [Evaluation Level] of AEROJOBS=30
85
   GOTO 89
86
   [Evaluation Level] of AEROJOBS=31
87
   GOTO 89
88
   [Evaluation Level] of AEROJOBS=32
89 If [Evaluation Level] of (AEROJOBS)    then . . .
   [Evaluation Level] of AEROJOBS=[Evaluation Level] of AEROJOBS
```

IT IS A MINI-PROGRAM WITHIN A PROGRAM. With a procedure you must first identify the field you want to work on or with. Then you identify what you want to do. You can add, subtract, multiply, divide, any field by any other field or number. **SELECTOR** calls numbers literals. You can also skip a step or a record. Then you can also delete, move, or goto a particular step. What you are doing is writing a program to manipulate data in any way you wish. I use these procedure steps in a number of ways. First, I use them to calculate fields which I had labeled as skip fields during data entry fields. Second, I use them to do table look-up and then do mathematics. That is something that you normally can not do with the typical data management system. There are illustrations of both as part of the article. One of those procedures checks a field in the original file, goes to a second file to get a value, places that value in the first file, looks up a field in a third file, gets a percentage from that field, multiplies the percentage by the number derived in the first action and stores that result in a separate field, adds .5 to that new number so that it will be rounded when it is truncated back into a field in the first file, looks up a third number field in the first file, goes to another file to get a value, places that value into a field in the first file, then adds the three fields in the first file and places that value in another field in the first file. WHEW.

Then, I run another procedure which takes all 89 procedure steps. I look up the value that was calculated by the first procedure to get a separate value. What I was doing was calculating salary grades by looking up a range of points, finding the grade, then placing that grade number into the first file. I was using grades 5 to 32. What I did was use the less than or equal to comparison. When the number fell into the lookup, I jump to another procedure step which gives the correct grade number, then jump to the end of the procedure so that the next record will be read. This procedure took me about one hour to design, test, debug, test, then do a production run. I defy other systems to be that responsive.

Now, remember those automatic modes I mentioned back at the start of this mass of words. I have gone through that procedure once, in order, and now just run a file called procedure and it actually does both passes through the data file then prints a series of three reports based on different sorts through that file I just created.

FORMATTED REPORT DEFINITIONS

The most sophisticated report writer I have found so far resides here. This module allows you to do anything you want with any piece of data. That's probably all I need to say. You can put data or labels anywhere you want, based on a set of up to five called files, and with the 89 procedure steps available to manipulate your data as you go from record to record. The procedure seems complicated the first time you use it. Then it becomes obvious to use and is one of the easiest I have used.

You should realize that this report generator is complicated, so it would probably be better to design your report on paper prior to starting to enter the schematic into the module.

```
B:SAMPLE.RPT FORMATTED REPORT DEFINITION as of JAN 28, ´83

Lines per page: 66.  Columns per page 125.

Files are: 1-AEROJOBS

Using 18/[Total Points] as Source Key Field

     Report Output Positioning of Literals and Variables

  Report Header Section
Line: 001%L"This is a sample REPORT HEADER"@C
Line: 003^D@C
Line: 010%L"SORTED BY TOTAL POINTS"@C
Line: 045^C@C
Line: 046^I@C
Line: 047^3@C
Line: 048^P@C

  Page Header Section
Line: 001%L"|POSITION    |POSITION                                  |KNOWLEDGE      | PRO +
       BLEM       |ACCOUNTABILITY| TOTAL | EVAL |  OLD  | COMP  |"@L1
Line: 002%L"|     CODE    | TITLE                                  | SKILLS        | SOL +
       VING       |                | POINTS| LEVEL|  LVL  | CODE  |"@L1
Line: 003%L"-"50@L1 %L"PAGE #  "@C ^I@S %L"-"50@L69

  Detail Line Section
Line: 004%L"|"@L1 %C1/1@R4 %C1/2@R9 %C1/3@R11 %L"|"@L13 %C1/5@L14 %L"|"@L44 +
       %C1/6@L46 %C1/7@R53 %C1/8@R58 %L"|"@L59 %C1/9@L61 %C1/10@R68 %C1/11@R +
       73 %L"|"@L74 %C1/12@L76 %C1/13@R83 %C1/14@R88 %L"|"@L89 %C1/18@R95 %L" +
       |"@L97 %C1/19@R103 %L"|"@L104 %C1/20@L110 %L"|"@L112 %C1/4@R118 %L"|"@ +
       L120
Line: 005%L" "@L1

  Page Footer Section
Line: 059%L" "@C
Line: 060%L"PRELIMINARY DATA NOT YET APPROVED BY INTEGRATION AND REVIEW COM +
       PANY"@C

  Report Footer Section
Line: 010%L"THIS IS THE FINAL PAGE OF A JOB EVALUATION MATRIX SORTED BY TOT +
       AL POINTS"@C
```

SELECTOR has a set of sections which stand alone. You may define a Report Header. This is a separate page for your report. Makes for a nice cover.

Then you may define a Record Header. This is a header which **SELECTOR** would print each time it reads a new record. Now, if you were creating invoices, you would probably change the record whenever the customer number changed. You would then create a entire, multi-line record header with customer name, address and ship to data as well as your letterhead, the invoice number and other goodies.

If you wish you may also define a separate page header. This is used if you are creating reports which would contain more than one page of data for each master record, or if you wished to create columnar reports within the formatted report module. **SELECTOR** also allows you to have a series of automatically incrementing numbers on both the record and page header sections. You can, therefore, let the program automatically assign invoice numbers, page numbers, or whatever, automatically. What happens is that the program queries you for all of the incrementing number starting numbers when you run the report.

Now you define the detail of the report. You define the report details in the same way which you define the page and record headers. You tell **SELECTOR** where you want the data placed, and what you want displayed. Here´s the shorthand which selector uses.

%N1/1@120 means that you wish to place the field name for field number one file number on in a left justified field which starts at line 20. %C1/1@R30 means that you wish to place the contents of field number one in file number one in a right justified field stating at row 30.

You can also float a $ sign in front of a numeric field, you can also concatenate two fields, you can center lines, you can assign line numbers, you can use printer control codes within each single line to take advantage of any special feature you have on your printer, and you can add literals (labels) to any place on the line you wish. You may also define as many lines as you wish and as long as you wish. That is some formatted report generator. There is an illustration of a formatted report using those job evaluation files as part of this article.

After all this, you still have other decisions to make. You may define a Page Footer if you wish. I have a small criticism here. You can not use the automatic numbering feature in the page footer. Only in the header sections can you sequence numbers. That means your page numbers must go at the top of the page, not on the bottom of the page. A small complaint, true, but with all the other advanced features, I wonder why this was not allowed?

Now, after your page footer, you may define a Record Footer. You get one record footer per record.

And, lastly, you may, if you wish, define a Report Footer at the end of your report. Don't laugh. This way you can have a cover sheet with an identifying label at the start and finish of your report. How many times have you picked up a report only to find you are at the end, not the beginning and wondered what you really had?

That is still not all. You may now define up to 89 procedure steps to be used in the processing of an individual record. That sure beats the 20 computed fields available with **DB MASTER.**

One other thing. Remember the name, address, phone number, and such which you defined before you called Micro-Ap so that you could unlock your system and get more than 60 records. Those things are now available as system variables so that you can insert them with just one keystroke into your report and no longer need to type them out.

If you wish to do sub-totals, grand-totals, and other manipulations with the formatted report generator, you would create a screen and record definition the same way that you would do if you were going to add real records to a specific data file, only you never add any records. Then, you use that data file definition as a called file when you create your report. You then use a procedure step to add the proper data to the proper fields. You would then print, page break, total or sub-total based on the information in that "dummy" file. How about that?

FILE TRANSLATION

The File Translation module allows you to read most non-**SELECTOR** data files into a **SELECTOR** data file or

```
B:AEROMATH.BCH BATCH POST TO FILE(S) MODULE as of JAN 28, '83

Files Used
1-AEROJOBS 2-POINTS 3-PERCENTS 4-RPOINTS


Calling & Called File & Field Names

Call#   Using: File#/Name - Field#/Name, Call: File#/Name - Field#/Name
  1:           1 AEROJOBS    7[KStep]          2 POINTS      1[Step]
  2:           1 AEROJOBS   10[PStep]          3 PERCENTS    1[Step]
  3:           1 AEROJOBS   13[RStep]          4 RPOINTS     1[Step]

BATCH POST TO FILE(S) MODULE Processing Procedures

 1 If [KStep] of (AEROJOBS) EQ [Step] of POINTS  then . . .
   [KPoints] of AEROJOBS=[Points] of POINTS
 2

 3 If [PStep] of (AEROJOBS) EQ [Step] of PERCENTS  then . . .
   [DUMMY] of PERCENTS=[KPoints] of AEROJOBS*[Percentage] of PERCENTS
 4
   [DUMMY] of PERCENTS=[DUMMY] of PERCENTS+.5
 5
   [PPoints] of AEROJOBS=[DUMMY] of PERCENTS
 6 If [RStep] of (AEROJOBS) EQ [Step] of RPOINTS  then . . .
   [RPoints] of AEROJOBS=[Points] of RPOINTS
 7
   [Total Points] of AEROJOBS=[KPoints] of AEROJOBS+[PPoints] of AEROJOBS
 8

 9
   [Total Points] of AEROJOBS=[Total Points] of AEROJOBS+[RPoints] of AEROJOBS
```

change a **SELECTOR** data file into another format. There are many programs which are set up to read and use **DBASE II** files. **SELECTOR** will create files for those programs to use. I use **ABSTAT** as a statistical package in the CP/M world. It reads those files with no problems. Also, if you have been using **DBASE II** and want to switch to **SELECTOR**, you can do so with very little effort. This also includes files created by or designed to be read by WordStar and that related set of programs.

REBUILD INDEX KEY FILE

Another of the main menu sections is a module which rebuilds your master key file, if you find it necessary. You will find that whenever you enter and edit records, this module is called upon and run automatically by **SELECTOR**. However, you might some time use **PIP** to concatenate files. Then you would use this item to create a correct key file for your use. I have also used it when I was doing data entry and editing and had to leave without exiting the program correctly. I shut it off rather than exited through the menu. What I had when I returned was a good data file with a old index file. I just rebuilt the **index** before I did anything else and all was fine.

FILE CONVERSION

One feature that any data base must have is the ability to restructure and create sub-sets of the file upon demand. You might want to split one large file in two. You may want to delete a field from everything. You may want to change the field lengths of a file. And on and on.

It works rather simply. You create your new screen and file definitions using that menu item. Then you create a file conversion definition by saying, on a filed by field basis, what you want in the new file and where to get that information from. It is very, very easy to use.

BATCH POST

I have saved the best and most useful to last. **SELECTOR** lets you post from one file to up to five other files at a time. I described two batch post definitions during the procedure steps discussions. And, since you can use that automatic run mode, you could actually post to a rather large set of files from one set of data input. You can build anything you want.

SUMMARY

SELECTOR seems to be the closest thing to duplicating the main frame environment that I have found for Micro Computers. You could also call this a "rave" review. I was able to find no bugs in the program. In fact, I liked it so much that I have switched all of my data base applications over to it.

If you need a data management system, I would have no hesitation in buying this one. Take some time to learn the variety of features and you will be happy with it. By the way, it is not a small program. The **APPLE** version comes spread on five disks. You would swap disks as you go from module to module.

CONTEMPORARY COMPUTING
P.O. BOX 4661
WALNUT CREEK, CALIFORNIA 94596

(415) 932-1921

24 HOUR NUMBER CALL ANYTIME

VERBATIM 5-1/4" minidiskettes

Datalife - 5 year warranty
SS, DD, RH - - - - - - - - - - - 10/$27.50
w/plastic box - - - - - - - - - - 10/$28.50

VEREX - specially made for APPLE
SS, DD, RH - - - - - - - - - - - 10/$24.95

QUENTIN RESEARCH

APPLE-Mate Disk Drive, 100% APPLE compatible
w/3' cable, plug into your controller card - $285.00

16K socketed RAM card - - - - - - - - - $79.00

Universal wire wrap card - - - - - - - - $35.00

FLIP-N-FILE - 50 diskette storage file - - - - - $21.95

FIBERBUILT - carrying case for APPLE w/2 drives
17x21x9 inches, removable cover - - - - $59.95

SEND CERTIFIED CHECK OR MONEY ORDER + 6.5 CAL SALES TAX + SHIP AND HANDLE UPS $2.50 PLUS 1% OF TOTAL ORDER

SF APPLE CORE
1515 Sloat Blvd., Suite 2
San Francisco, CA. 94132

BULK RATE
U.S. POSTAGE
PAID
Permit No. 290
Concord, CA 94520

TO:

3121AUG83
DOUGLAS STORKOVICH
921 PINTO CT
WALNUT CREEK CA 94596

A Review of the '80 Columbards